Paternotte Mattéo

TP°2

Exercice 2 :

Calcul es intégrales :

$\int_0^2 t^2 dt = \frac{8}{3}$ $\int_0^1 t^2 dt = \frac{1}{3}$ $\int_0^2 e^t dt = e^2 - 1$ $\int_0^1 e^t dt = e^1 - 1$

Programme rectangle :

```
%approximation par la méthode des rectangles de l'intégrale de ft)=t^2 sur
%[0,2], I=8/3
clear all
format short
I=input('entrer I (valeur exacte de l integrale )->');
a=input('entrer a (borne inf)->');
b=input('enttrée b (borne sup)->');
n=input('input nombre intervalle (n) ->');
h=(b-a)/n;
fa=(a^2);%f(a)
fb=(b^2);%f(b)
%formule des rectangle
ff=0;
for i=1:n-1
x=a+(i*h);
fx=x^2;
ff=ff+fx;%somme de f(a+ih) où i=1 à n
end
Rn=h*(ff+ fa)%résultat de l'intégration
erreur =abs(I-Rn)%erreur
```

Programme trapèze :

```
%approximation par la méthode des trapézes l'intégrale de f(t)=t^2 sur
%[a,b]
clear all;
format short ;
I=input('entrer I (valeur exacte de l integrale )->');
%entré des valeurs , initialisation de la boucle
a=input('entrer a (borne inf)->');
b=input('enttrée b (borne sup)->');
n=input('input nombre intervalle (n) ->');
h=(b-a)/n;
fa=(a^2);%f(a)
fb=(b^2);%f(b)
%formule des trapezes
ff=0;
for i=2:n
    x=a+h*(i-1);
    fx=x^2;
    ff=ff+2*fx;% somme de 2*f(a+i(h)) où i=1 à n-1
end
Tn=(h/2)*(fa+fb+ff)%résultat de l'approximation
En=abs(I-Tn)% erreur
```

Programme Simpson :

```
%approximation par la méthode des trapézes l'intégrale de f(t)=t^2 sur
%[a,b]
clear all;
format short ;
I=input('entrer I (valeur exacte de l integrale )->');
%entré des valeurs , initialisation de la boucle
a=input('entrer a (borne inf)->');
b=input('enttrée b (borne sup)->');
n=input('input nombre intervalle (n) ->');
h=(b-a)/n;
fa=(a^2);%f(a)
fb=(b^2);%f(b)
ff=0;
%tous les termes 4*f(a+nh) termes a f(b)
%h=(1,3,5,7,...,n-1)
for i=2:2:n;
    x=(a+(i-1)*h);
    fx=(x^2);
    ff=ff+4*fx;
end
%tous les termes 2*f(a+nh) à f(b)%h=(2,3,4,6,...,n-2)
for i=3:2:n;
    x=(a+(i-1)*h);
    fx=(x^2);
    ff=ff+2*fx;
end
Sn=(h/3)*(fa+fb+ff)%résultat de l'approximation
En=abs(I-Sn)% erreur
```

Valeurs calculées	[0,2]	[0,1]	I_2 [0,2]	I_2 [0,1]
Rn	2.2800	0.285	5.7714	1.6338
Erreur rectangle	0.3867	0.0483	0.6177	0.0845
Tn	2.6800	0.335	6.4103	1.7197
Erreur trapèze	0.0133	0.0017	0.0212	0.0014
Sn	2.6667	0.3333	6.3891	1.7183
Erreur Simpson	0	0	1.2621e-05	$1.72*10^{-5}$

Exercice 3 :

31.

```
function y =fun(x)
y=x.^2;


clear all
Rn=rectangle(@fun, 0, 2, 10)


%fonction permettant l'approximation d'une intégrale par la ùéthode des
%rectangles . En Entrée, l'expression de la fonction à intégrer, les bornes
%a et b, le nombre de pas n. La fonction élaborée renvoie l'approximation
%Rn.

function Rn=rectangle(fun, a, b, n)
    h=(b-a)/n;
    fa=fun(a);%f(a)
    vb=fun(b);%f(b)
    %méthode des rectangles
    ff=0;
    for i=1:n-1
    x=(a+(i*h));
    fx=fun(x);
    ff=ff+fx;%somme de f(a+i*h) où i=1 à n
end
Rn=h*(ff+fa);%résultat de l'intégration
```

Valeurs calculées	[0,2]
Rn	2.28

32.

```
clear all
Rn=rectangle(@fun, 0, 2, 10)
Tn=trapeze(@fun, 0, 2, 10)
Sn=simpson(@fun, 0, 2, 10)


function Tn=trapeze(fun, a, b, n)
h=(b-a)/n;
fa=fun(a);%f(a)
fb=fun(b);%f(b)
%formule des trapezes
ff=0;
for i=2:n
    x=a+h*(i-1);
    fx=fun(x);
```

```
    ff=ff+2*fx;% somme de 2*f(a+i(h)) où i=1 à n-1
end
Tn=(h/2)*(fa+fb+ff);
```

```
function Sn=simpson(fun, a, b, n)
h=(b-a)/n;
fa=fun(a);%f(a)
fb=fun(b);%f(b)
ff=0;
%tous les termes 4*f(a+nh) termes a f(b)
%h=(1,3,5,7,...,n-1)
for i=2:2:n
    x=(a+(i-1)*h);
    fx=fun(x);
    ff=ff+4*fx;
end
%tous les termes 2*f(a+nh) à f(b)%h=(2,3,4,6,...,n-2)
for i=3:2:n
    x=(a+(i-1)*h);
    fx=fun(x);
    ff=ff+2*fx;
end
Sn=(h/3)*(fa+fb+ff);%résultat de l'approximation
```

Valeurs calculées	[0,2]
Tn	2.6800
Sn	2.6667

33.

```
clear all
bb=0;
bh=pi/2;
n=1000;
I=1;
rectangle(@fun, bb, bh, n, I)
trapeze(@fun, bb, bh, n, I)
simpson(@fun, bb, bh, n, I)
```

```
function Rn=rectangle(fun, a, b, n, I)
    h=(b-a)/n;
    fa=fun(a);%f(a)
    fb=fun(b);%f(b)
    %méthode des rectangles
    ff=0;
    for i=1:n-1
    x=(a+(i*h));
    fx=fun(x);
    ff=ff+fx;%somme de f(a+i*h) où i=1 à n
end
Rn=h*(ff+fa);%résultat de l'intégration
erreur =abs(I-Rn);%erreur
```

```
function Tn=trapeze(fun, a, b, n, I)
h=(b-a)/n;
fa=fun(a);%f(a)
fb=fun(b);%f(b)
%formule des trapezes
ff=0;
for i=2:n
    x=a+h*(i-1);
    fx=fun(x);
    ff=ff+2*fx;% somme de 2*f(a+i(h)) où i=1 à n-1
end
Tn=(h/2)*(fa+fb+ff);
En=abs(I-Tn)% erreur
```

```
function Sn=simpson(fun, a, b, n, I)
h=(b-a)/n;
fa=fun(a);%f(a)
fb=fun(b);%f(b)
ff=0;
%tous les termes 4*f(a+nh) termes a f(b)
%h=(1,3,5,7,...,n-1)
for i=2:2:n
    x=(a+(i-1)*h);
    fx=fun(x);
    ff=ff+4*fx;
end
%tous les termes 2*f(a+nh) à f(b)%h=(2,3,4,6,...,n-2)
for i=3:2:n
    x=(a+(i-1)*h);
    fx=fun(x);
    ff=ff+2*fx;
end
Sn=(h/3)*(fa+fb+ff);%résultat de l'approximation
En=abs(I-Sn)% erreur
```

```
function y =fun(x)
y=exp(x);
```

```
function y =fun(x)
y=cos(x);
```

Valeurs calculées	$\int_0^{\pi/2} \cos(t)\, dt$	$\int_{-3}^{3} e^t dt$
Valeur réelle de l'intégrale	1	$e^3 - e^{-3} = 20.0357$
Rn	1.0765	3.5190
Er	0.0765	16.5167
Tn	0.9979	3.8183
Et	0.0021	16.2174
Sn	1.0000	3.8105
Es	3.3922e-06	16.2252

34.

```
clear all
bb=0;
bh=pi/2;
n=10;
I=1;
rectangle(@fun, bb, bh, n, I)
trapeze(@fun, bb, bh, n, I)
simpson(@fun, bb, bh, n, I)


function Rn=rectangle(fun, a, b, n, I)
    h=(b-a)/n;
    fa=fun(a);%f(a)
    fb=fun(b);%f(b)
    %méthode des rectangles
    ff=0;
    for i=1:n-1
    x=(a+(i*h));
    fx=fun(x);
    ff=ff+fx;%somme de f(a+i*h) où i=1 à n
end
Rn=h*(ff+fa);%résultat de l'intégration
Er =abs(I-Rn);%erreur


function Tn=trapeze(fun, a, b, n, I)
h=(b-a)/n;
fa=fun(a);%f(a)
fb=fun(b);%f(b)
%formule des trapezes
ff=0;
for i=2:n
    x=a+h*(i-1);
    fx=fun(x);
    ff=ff+2*fx;% somme de 2*f(a+i(h)) où i=1 à n-1
end
Tn=(h/2)*(fa+fb+ff);
Et=abs(I-Tn)% erreur


function Sn=simpson(fun, a, b, n, I)
h=(b-a)/n;
fa=fun(a);%f(a)
fb=fun(b);%f(b)
ff=0;
%tous les termes 4*f(a+nh) termes a f(b)
%h=(1,3,5,7,...,n-1)
for i=2:2:n
    x=(a+(i-1)*h);
    fx=fun(x);
    ff=ff+4*fx;
end
%tous les termes 2*f(a+nh) à f(b)%h=(2,3,4,6,...,n-2)
for i=3:2:n
    x=(a+(i-1)*h);
    fx=fun(x);
```

```
    ff=ff+2*fx;
end
Sn=(h/3)*(fa+fb+ff);%résultat de l'approximation
Es=abs(I-Sn)% erreur
```

Méthode	n=10	n=100	n=1000
Erreur Rectangle	0.0765	0.0078	7.85e-04
Erreur Trapèze	0.0021	2.0562e-05	2.06e-7
Erreur Simpson	3.3922e-06	3.3824e-10	3.26e-14

Exercice 4 :

411.

Si x=2

```
function f= fun(x)
f=x^2
f=4

function g=dana(x)
g=2*x
g=4
```

412.

a)

```
%%derive a droite
clear;clc
format short ;
h=0.1
i=1
for x=-10:h:10;
    df=(fun(x+h)-fun(x))/h
    Yd(i)=df
    i=i+1
end
```

```
Pour x=-10 et 10
```

f =

 102.0100

f =

 100

df =

 20.1000

Yd=

-19.900 to 20.100

i =

 202

b)

```
%% derive à gauche
clear;clc
format short ;
h=0.1
i=1
for x=-10:h:10;
    df=(fun(x)-fun(x-h))/h
    Yg(i)=df
    i=i+1
end
```

```
Pour x=-10 et 10
```

f =

100

f =

98.0100

df =

19.9000

Yg=

-20.100 to 19.900

i=

202

c)

```
%% derive symetrique
clear;clc
format short ;
h=0.1
i=1
for x=-10:h:10;
    df=(fun(x+h)-fun(x-h))/(2*h)
    Ys(i)=df
    i=i+1
end
```

```
Pour x=-10 et 10
```

f =

102.0100

f =

98.0100

df =

20.0000

Ys =

-20.000 to 20.000

i =

202

d)

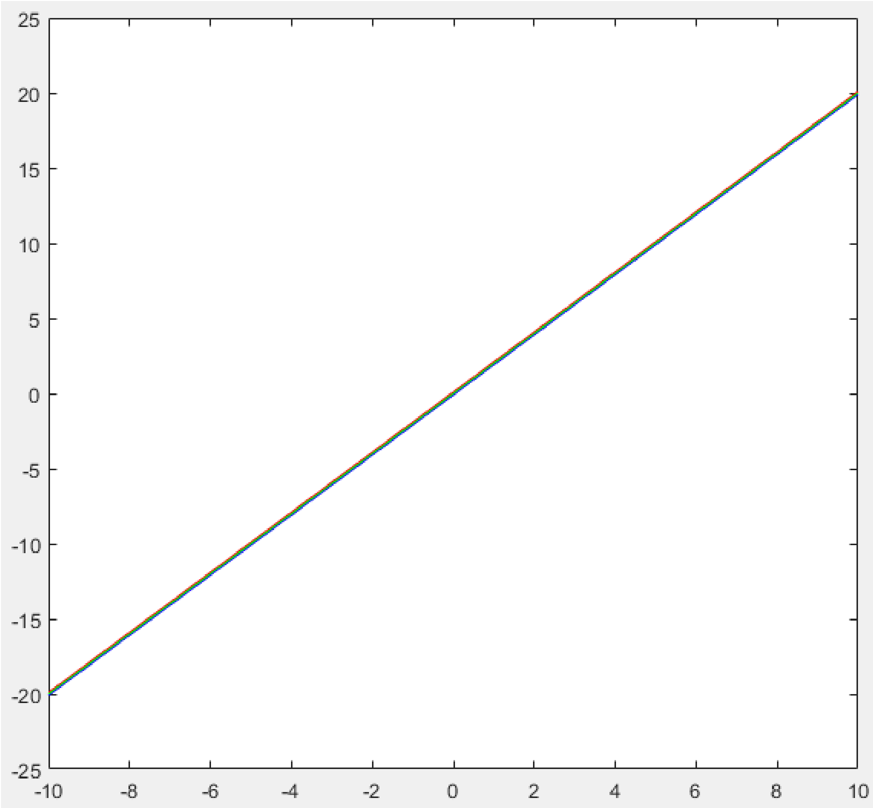
```
%derive a droite
clear;clc
format short ;
h=0.1;
i=1;
for x=-10:h:10
    df=(fun(x+h)-fun(x))/h;
    Yd(i)=df;
    i=i+1;
end

% derive à gauche
clear;clc
format short ;
h=0.1
i=1
for x=-10:h:10;
    df=(fun(x)-fun(x-h))/h
    Yg(i)=df;
    i=i+1
end

% derive symetrique
clear;clc
format short ;
h=0.1
i=1
for x=-10:h:10;
    df=(fun(x+h)-fun(x-h))/(2*h)
    Ys(i)=df
    i=i+1
end
```

```
x=-10:h:10
figure(1);
plot(x,Yd,'r');
hold on ;
plot(x,Yg,'b');
hold on ;
plot(x,Ys,'g');
hold off ;
```

On peut remarquer que cela correspond à nos résultats précédents.



413.

a)

```
%% calcul de la derive analytique g(x)
clear;clc
format short ;
h=0.1
i=1
for x=-10:h:10;
    df=dana(x)%calcul de la derive analytique
    Ya(i)=df %tableau des derivees analytique
    i=i+1
end
```

Pour x=-10 et 10

g =

20

df =

20

Ya =

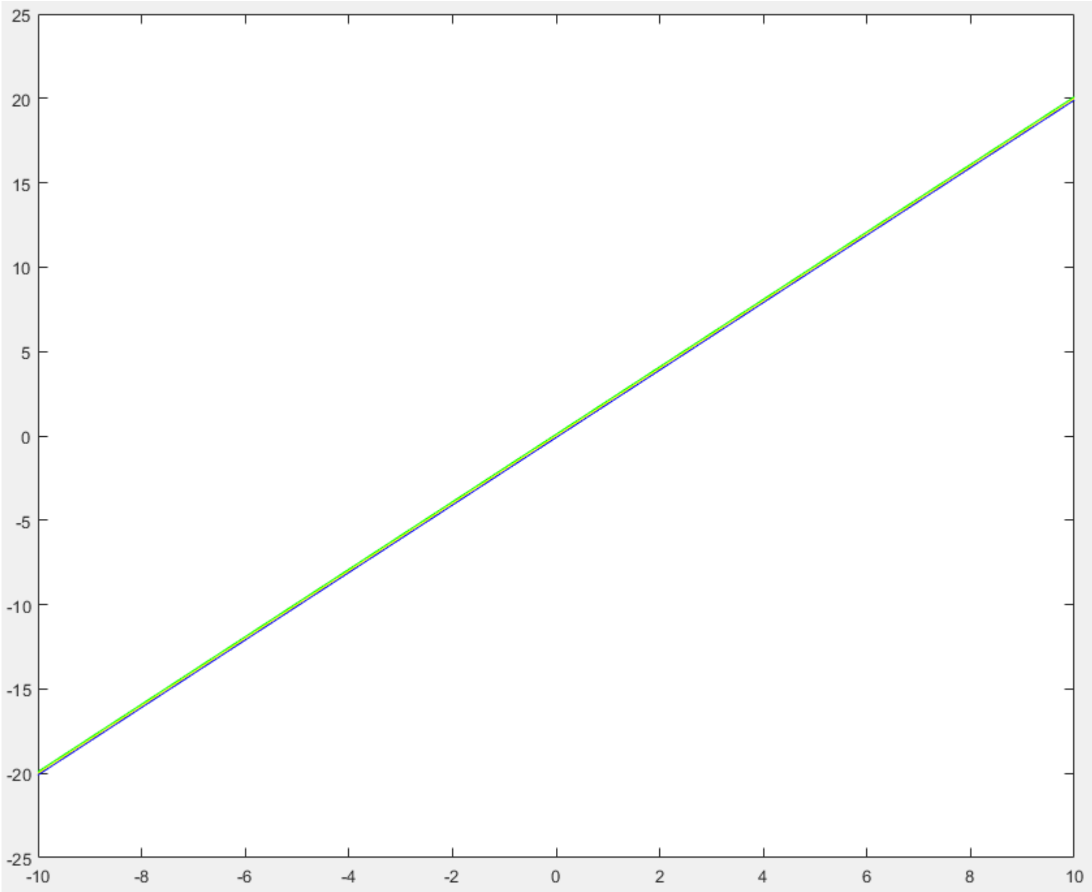
-20.000 to 20.000

i =

202

B)

```
figure(1);
plot(x,Ya,'r');
hold on ;
plot(x,Yd,'g');
hold on ;
plot(x,Yg,'b');
hold on ;
plot(x,Ys,'y');
hold off ;
```



c)

```
Ed=sum(abs(Ya-Yd))/length(x)
Eg=sum(abs(Ya-Yg))/length(x)
Es=sum(abs(Ya-Ys))/length(x)
Erreur=[Ed Eg Es]


Ed =
    0.1000
Eg =
    0.1000
Es =
   3.1286e-14
Erreur =
    0.1000    0.1000    0.0000
```

42.
421.
a)

```
function f= fun2(x)
f=x^2*exp(cos(x))
fun2(2)
```

```
f =
     2.6383

function g =dana2(x)
g=2*x*exp(cos(x))+x^2*(-sin(x))*exp(cos(x))

dana2(2)
g =
     0.2393
```

b)

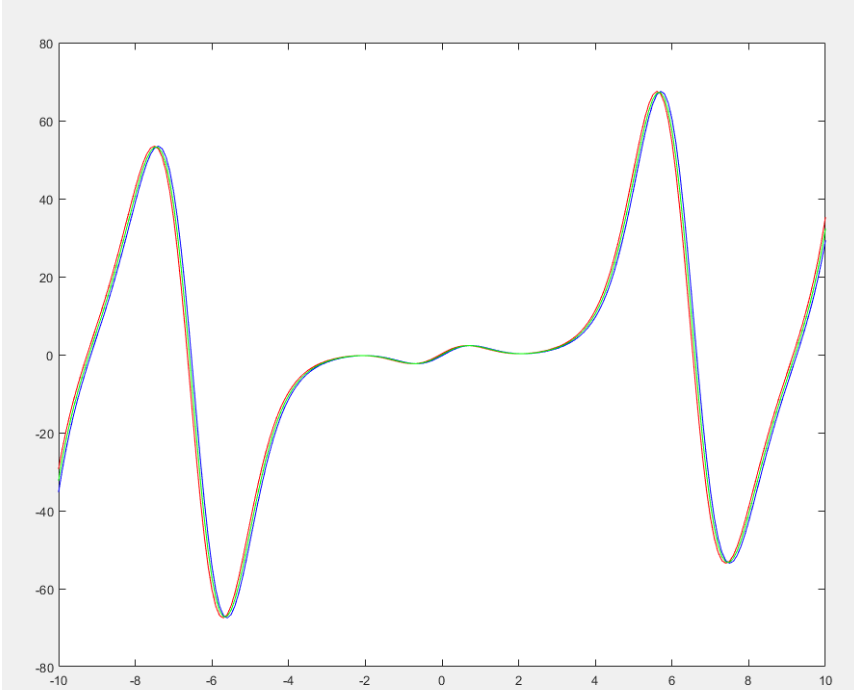
```
%derive a droite
format short ;
h=0.1;
i=1;
for x=-10:h:10
    df=(fun2(x+h)-fun2(x))/h
    Yd(i)=df
    i=i+1
end

% derive à gauche
format short ;
h=0.1;
i=1;
for x=-10:h:10
    df=(fun2(x)-fun2(x-h))/h
    Yg(i)=df
    i=i+1
end

% derive symetrique
format short ;
h=0.1;
i=1;
for x=-10:h:10
    df=(fun2(x+h)-fun2(x-h))/(2*h)
    Ys(i)=df
    i=i+1
end
```

c)

```
x=-10:h:10
figure(1);
plot(x,Yd,'r');
hold on ;
plot(x,Yg,'b');
hold on ;
plot(x,Ys,'g');
hold off ;
```



d)

```
%calcul de la derive analytique g(x)
format short ;
h=0.1
i=1
for x=-10:h:10;
    df=dana2(x)%calcul de la derive analytique
    Ya(i)=df %tableau des derivees analytique
    i=i+1
end

Ed=sum(abs(Ya-Yd))/length(x)
Eg=sum(abs(Ya-Yg))/length(x)
Es=sum(abs(Ya-Ys))/length(x)
Erreur=[Ed Eg Es]
```

422.

Pour les erreurs max :

```
Ed=max(abs(Ya-Yd));
Eg=max(abs(Ya-Yg));
Es=max(abs(Ya-Ys));
Erreur=[Ed Eg Es]
```

h	0.1	0.05	0.01

Erreur Droite	5.6864	2.8479	0.5698
Erreur Gauche	5.6864	2.8479	0.5698
Erreur Symétrique	0.3069	0.0769	0.0031

Les erreurs diminuent plus h est petit donc si on veut de plus en plus faible erreur on doit continuer à baisser h.